

## FLEXIBLE EFFICIENT BRANCHING PARTICLE TRACKING ALGORITHMS

10 Be it known that we, Michael A. Kouritzin, a citizen of Canada, and Klaus E. Fleischmann, a citizen of Germany, have invented new and useful improvements in the above-mentioned invention, of which the following is a specification.

15

## BACKGROUND OF THE INVENTION

The present invention relates to a method and a system for efficient non-linear optimal tracking, smoothing, and prediction of dynamic processes in real time

20 Many industries require a system to determine the location and state of a signal in some domain. For example, one might want to know the position, orientation, and velocity of an aircraft over a busy airport. Additionally, it might be important to know where the aircraft is likely to be in the future and what path it has taken in the past. In many such cases full information about the signal is not available. Instead, only distorted, corrupted, partial information given by discrete-time observations is provided. In the aircraft example, the only data provided to the airport might be noisy transmissions of truncated altimeter and range data from the aircraft that arrive at two second intervals. While perfect determination of the signal state is impossible under these circumstances, it is still desirable to obtain probabilistic estimates of the state conditioned on the information that is available. Preferably one should also obtain smoothed estimates of past signal states, probabilistic information on the likelihood of various signal paths, and predictions of future states given all observations up to some time.

35 More precisely stated, the problem is as follows. The signal  $X_t$  to be tracked is a cadlag Markov process, for example it may follow an Ito equation

$$dX_t = A(X_t) + B(X_t)dB_t,$$

5 or more generally a Martingale problem on a complete separable metric space. The discrete observations  $Y_{t_k}$ , occurring at times  $t_k$ , are dependent on the signal in a stochastic manner described by

$$Y_{t_k} = h_k(X_{t_k}, V_k),$$

10 where  $V_k$  is a sequence of random variables that model the noise input into this time-varying dependence. These formulae might not be precisely known in a particular instance and some extra randomness can be introduced to model this implementation uncertainty.

The requirement is to construct a device which can efficiently determine the conditional distribution of the state of the signal given all observations up to the current time, that is,

$$P(X_{t_k} \in dx | Y_{t_i}, i \leq k).$$

20 This device is known as an optimal tracker. Also, the device should provide smoothers to estimate past signal states and predictors to estimate future signal states, that is,

$$P(X_{\tau_s} \in dx | Y_{t_i}, i \leq k)$$

and

$$P(X_{\tau_p} \in dx | Y_{t_i}, i \leq k),$$

where  $\tau_s < t_k$  and  $\tau_p > t_k$ . Indeed, one is often interested in

$$P(X_{[\tau_s, \tau_p]} \in dz | Y_{t_i}, i \leq k),$$

25 where  $z$  ranges over the cadlag functions and  $X_{[\tau_s, \tau_p]}$  represents the path of the signal between times  $\tau_s$  and  $\tau_p$ .

In the specific Ito equation case in which the stochastic dynamics given by  $A$  and  $B$  are linear, the observation function  $h$  is of the form

$$h_k(X_{t_k}, V_k) = \hat{h}_k X_{t_k} + V_k \text{ (additive noise),}$$

30 and each  $V_k$  is an independently distributed Gaussian random variable, an efficient recursive optimal tracker is already known: the Kalman filter. Recursive, in this context, means that the computation involved with each new observation requires the data of that observation only with no direct reference to the data of any previous observation. This allows the Kalman filter to operate

35 very quickly. However, as noted, the Kalman filter has very stringent

5 requirements on the possible signal dynamics and on the form of the observation noise. As well, it provides no direct method to provide predictors or smoothers.

Despite these shortcomings, the Kalman filter is the core of most current devices to provide filtering solutions. In linear Gaussian additive environments, trackers based on the Kalman filter produce optimal estimates. For more general environments the Kalman filter is suboptimal, but can still provide some solution in the form of the extended Kalman filter. Later developments have included running multiple Kalman filters in banks, each with different assumptions about the signal dynamics, and weighting the Kalman filters and their inputs to provide a track. This method is called the interacting multiple model tracker, and provides improved but still suboptimal tracking.

New techniques have expanded the class of filtering problems that can be solved using exact methods. These techniques provide readily implementable computer algorithms that are nearly as efficient as the Kalman filter and provide optimal solutions. However, the problems that can be solved in this manner still form a small subset of the whole space of real world problems.

A more recent type of general solution to the filtering problem is the set of particle methods. These techniques simulate a large number of independent samples  $\xi_t^j$  of the signal, called particles. The set of particles  $\xi_t^j, 0 \leq j \leq N$  are then adjusted in some manner to conform to each observation, possibly by assigning each a scalar weight value  $W_t^j$  or by interchanging the state values associated with each particle in some manner. The conditional distribution of the state of the signal at the current time is approximated by the locations of the particles and their weights, specifically,

$$P(X_{t_k} \in S | Y_{t_i}, i \leq k) \approx \frac{1}{\sum_{j=1}^N W_{t_k}^j} \sum_{j=1}^N W_{t_k}^j \delta_{\xi_{t_k}^j}(S).$$

The particle adjustment is performed in such a manner that the resulting weighted particle distribution converges to the optimal conditional distribution of the signal, given the observations, as the number of particles,  $N$ , is increased. Predicted distributions for some time  $\tau_p > t_k$  conditioned on ob-

5   servations up to time  $t_k$  are obtained by simulating the particles forward in time without any adjustment.

Two possible particle methods are described in U. S. Patent 5,933,352 published on Aug. 3, 1999 in the name of Gerard Salut, entitled "Method and a system for non-linear optimal estimation of dynamic processes in real time". In the first system, the claimed method of adjusting the particles to match the observations is solely to maintain a weight value associated with each particle and to modify this weight in accordance with each observation. This method has two sub-methods, in one of which the influence of past observations on the particle weights is attenuated in time, and in the other of which the influence of past observation on the weights is strictly limited in time.

A second system claimed by U. S. Patent 5,933,352 is to adjust the particles after each observation by redistributing some or all of their state data. This random redistribution is conducted such that particles with a higher weight are more likely to have their state data overwrite the state data of other particles that are part of the redistribution process. In this manner, the particle states are preferentially collected around the areas of the state space which have the highest conditional probability of containing the signal. While U.S. Patent 5,933,352 does not specifically state this, it can be assumed that the intention is for all particles that are involved in this redistribution to have their weights set to the average of the weights of all involved particles after the redistribution is complete, since this is the condition which will retain the asymptotic optimality of the system.

As U. S. Patent 5,933,352 states, simple particle weighting without redistribution causes degeneration in the output conditional distribution that can be rectified only to some degree by attenuating or limiting the effect of older observations. These actions themselves entail an unavoidable loss of asymptotic optimality in increasing numbers of particles for the filter. If only some particles are redistributed, there are still some degenerative effects working against optimality, and the calculation of the average weight for a subset of the particles is cumbersome.

5           However, a great amount of computation is involved in redistributing all  
of the particle states at each time step. This effort is especially pronounced if  
the particle data must travel between processing units in a distributed  
architecture. Moreover, redistributing all of the particles instead of merely  
10   considering each particle to determine whether redistribution or branching (as  
discussed below) is appropriate for it induces undue extra randomness that  
impairs path space capabilities and downgrades performance.

          Neither of the systems claimed in U. S. Patent 5,933,352 provide  
smoothing distributions, that is, estimates of the signal state for times  $\tau_s < t_k$ ,  
given the observations up to time  $t_k$ . They also provide no estimates of the  
15   historical path of the signal.

#### BRIEF SUMMARY OF THE INVENTION

20           The present invention relates to a method and a system for efficient non-  
linear optimal tracking, smoothing, and prediction of dynamic processes in real  
time. The method used is to store a large number of independent samples of  
the signal state, known as particles. The branching particle-based nonlinear  
filter accepts a sequence of measurements from sensors, each of which contain  
noisy, corrupted, and distorted information about the position of the signal (i.e.  
"measurement noise"), adjusts its particles in a novel manner such that they  
25   conform to the measurements, and then uses these particles to provide  
smoothing, tracking, and predicting approximate conditional distributions of  
the signal state. These distributions can be used to provide a human display  
interface or to control automated systems.

#### BRIEF DESCRIPTION OF THE DRAWINGS

30           The method of the invention is illustrated by the drawings in which:

Figure 1 shows the filtering process;

Figure 2 shows the filter observation process;

Figure 3 shows the evolve particle process;

35           Figure 4 shows the branch particle process;

Figure 5 shows the renormalize particle count process;  
 Figure 6 shows the duplicate particle process, and  
 Figure 7 shows the remove particle process.

## DETAILED DESCRIPTION OF THE INVENTION

In accordance with the method of the invention, a filter is shown in Figure 1. In processing each observation (Figure 2: Filter Observation), the filter first evolves each particle state forward in time (Figure 3: Evolve Particle) and removes (Figure 7: Remove Particle) particles that have left the bounds of the signal domain, if the particle evolution allows this. Each particle is then branched (Figure 4: Branch Particle) according to the observation, and after each particle has been processed, the total particle count is renormalized (Figure 5: Renormalize Particle Count).

To evolve a particle (Figure 3: Evolve Particle), the particle state is first copied to a separate logical location in the device, and the original is marked as the direct ancestor of the new particle. A forward evolution of the state of the new particle is then simulated according to the stochastic law of the signal process, or if that is not available then some perturbed approximate model of the signal process. The time period for this forward evolution is given by the length of time between the preceding observation and the observation presently under consideration.

After evolving each particle, the particles are then branched (Figure 4: Branch Particle) to account for the information from the observation. A value labeled  $\xi_k(x)$  is calculated for a given particle  $x$  in response to the observation at time  $t_k$  according to the formula

$$\xi_k(x) = \min\{|\zeta_k(x)|, 1\} \cdot \text{sgn}\{\zeta_k(x)\},$$

where

$$\zeta_k(x) = \frac{\theta_k(g_k(Y_k, x))J_k(Y_k, x)}{\theta_k(Y_k)} - 1.$$

In this formula,  $\theta_k$  is the density function for the random variable  $V_k$  that models the noise input, the function  $g_k(Y_k, x) = V_k$  is the inverse map of

5 the function  $h_k(x, V_k) = Y_k$  that describes the observations, and  $J_k(Y_k, x)$  is the determinant of the matrix  $\{\frac{\partial g_k^i}{\partial Y_k^j}(Y_k, x)\}_{i,j=1}^m$ . These formulae are constructed specifically so that the branching method will provably converge to the optimal filter as the number of particles is increased.

For additive zero mean,  $\sigma^2$  variance Gaussian noise such as the  
10 Kalman filter requires, the formula for  $\zeta_k$  becomes

$$\zeta_k(x) = \frac{\exp\{h_k(x) \cdot Y_k - 2\sigma^2 \|h_k(x)\|^2\}}{\sigma^2} - 1$$

Similarly to incompletely understood signal dynamics, the model of the noise which is used to form  $\theta_k$  and  $g_k(Y_k, x)$  can be approximate if no better  
15 knowledge is available.

A uniform-(0,1) random variable  $U$  is generated for the particle to be branched, and a comparison is then made between this  $U$  and  $\xi_k(x)$ .  $U \geq |\xi_k(x)|$  then the particle is not affected by the branching. This is common, especially (because of the form of the calculation of  $\xi_k(x)$ ) with more difficult  
20 types of noise that most reduce the information contained in an observation.

If  $U < |\xi_k(x)|$  then, in the case that  $\xi_k(x) > 0$ , the particle is duplicated (Figure 6: Duplicate Particle), and in the case that  $\xi_k(x) < 0$ , the particle is removed (Figure 7: Remove Particle).

To re-normalize the total particle count (Figure 5: Re-normalize Particle  
25 Count), a value  $l$  is set to equal the difference between the current number of particles and the original number. If the current number of particles is greater than the original number, then  $l$  of those particles are selected at random to be removed (Figure 7: Remove Particle). If the current number of particles is less than the original number, then if this number is less than half of the original  
30 number of particles, all particles are duplicated (Figure 6: Duplicate Particle), updating the value of  $l$  each time, until the current number of particles is within half of the original number, and then  $l$  of the particles are selected at random to be duplicated again. At the end of either possibility the current total number of particles is equal to the original number.

5 Duplicating a particle (Figure 6: Duplicate Particle) is performed by first copying the particle state to a separate logical location in the device. The count of the number of particles is then incremented, the new particle is marked as having the same direct ancestor as the original, and the weight value of all ancestor particles (chaining back recursively to include the marked ancestors  
10 of ancestor particles) is incremented by one.

To remove a particle (Figure 7: Remove Particle), first delete the particle state data and decrement the count of the number of particles. Then decrement the weight of all ancestors (chaining back recursively as described above) of the  
15 deleted particle, and if the weight of any of these ancestors is reduced to zero, then delete that ancestor particle.

As is the case for any particle filter, the particle locations provide the information to construct the approximated conditional distribution of the signal state. For the optimal tracking filter, the current particles are used with weight  
20 value of one for each. To construct an optimal predicting filter, evolve a copy of the current particles forward to the time for which the prediction is to occur, and use a weight value of one for each.

This new branching particle method also allows the construction of optimal smoothing filters. The purpose of the ancestor particles is to retain probabilistic data about the likely historical path of the signal. To use these  
25 particles to form a smoothing filter for time  $\tau_s < t_k$ , select the set of ancestor particles that were created at time  $t_{k_s}$ , where  $t_{k_s} \leq \tau_s$  but is the most recent observation time for which this is true. Copy each of these ancestor particles and evolve them forward to the time  $\tau_s$ , if  $\tau_s \neq t_{k_s}$ . Then these particles, weighted by their associated ancestor particle weights, provide the approximate  
30 conditional distribution of the signal state at time  $\tau_s$ . If it is known in advance for which times smoothing filters will be required, then the state of each particle at that time in its evolution can be stored as ancestor particles as well, and no forward evolution will be required during the construction of a smoothing filter. The weight for each of these intermediate ancestor particles  
35 will be the same as that of their direct ancestor. Ancestor particle data



5 older than a certain amount of time can be discarded in order to bound the resource usage without disrupting estimates within the remaining period.

The branching particle filter operates recursively on the observation data, allowing real-time operation of the system. It is asymptotically optimal in increasing numbers of particles and in a decreasing period of time between  
10 observations, but the rate of convergence with regards to the observation period is extremely fast, so that the difference between the approximation and the optimal filter for practical observation periods is essentially dependent on the number of particles. The particle data can easily be distributed among multiple hardware systems such that operations are efficiently computed in  
15 parallel, allowing large numbers of particles to be used practically. In a distributed system, re-normalization of the particle count can be conducted among each of the hardware units separately, or a more complicated scheme including particle transfer among hardware units can be implemented if data transfer within the system is rapid.

A major improvement of our branching particle filter over other methods  
20 is the cautious branching, by which is meant the high probability that particle branching will not affect a particle rather than duplicating or removing it. This aspect of the method allows a considerable increase in the speed of the filter, which can be traded for an increase in the number of particles within the same amount of processing power, and thus a closer approximation to the optimal  
25 filter. Even at the same particle count as other particle filters, the branching particle filter evinces superior performance in high observation noise cases since there is a smaller variance induced by particle re-distributions. This means that there is a smaller probability that the particles will be redistributed  
30 to erroneous locations. For example, the particles may be less likely to redistribute themselves to a single or a few selected previous particle sites in a too focused manner and erroneously attribute too great a commitment to an incorrect determination of the signal state. These advantages are more apparent as the observations become more rapid or as the observation noise  
35 becomes more extensive, both of which are situations that increase the

5 practical difficulty of the filtering problem and for which good solutions are  
most useful, since in these cases the particles are even less likely to be affected  
during a branch. In situations of extreme observation rate and noise distortion,  
it is possible with some noise models to approximate and shorten the  
calculation of the  $\sim k$  value without disrupting the asymptotic optimality of the  
10 filter, and thereby to decrease the calculation time for each particle further.

The cautious branching is not merely an alternate, faster, and more  
accurate method for approximating the optimal filter; it is a device which allows  
the storage of information required to provide optimal smoothers and full  
historical path filtering by ensuring that past state information is properly  
15 associated with each particle. The method of U. S. Patent 5,933,352 and other  
particle filters destroys this information by redistributing particles to one of a  
randomly selected set of states at each observation period, disregarding the  
path that the particle took in the state space to arrive at this new location. The  
availability of this full historical path data allows the system to provide an even  
20 larger set of information than just optimal smoothers. For example, the particle  
system can be queried to determine the probability that the signal moved from  
one location to another through a particular path at a particular time. This is a  
major advancement in the capabilities of practical, general purpose nonlinear  
filters.